Large scale semantic representation with flame graphs

Alastair BUTLER Kei YOSHIMOTO
Institute for Excellence in Higher Education, Tohoku University

1 Introduction

Flame graphs are a recent data representation technique (Gregg 2013) applied to making informative large volumes of information from stack traces listing executed functions of a computer system. This paper showcases flame graphs as a way to concisely present information calculated when there is the formal semantic analysis of natural language (see e.g., Dowty, Wall and Peters 1981). A convenient visual overview of large quantities of semantic data is offered as well as the ability to zoom-in on particulars concerning entities (mentioned people, objects, etc.), predicate relations (properties of entities, what entities do, etc.), argument roles (anchoring entities to predicate relations), connectives (establishing sentence and discourse content), as well as operators with scope (such as negation to invert or modality to qualify meaning).

Succinct presentation has value with the growth of large scale semantically annotated corpora (assembled corpus resources include Abstract Meaning Representations (Banarescu et al 2013), Deepbank (Flickinger et al 2012), Groningen Meaning Bank (Basile et al 2012), Treebank Semantics Corpus (Butler and Yoshimoto 2012) and Universal Conceptual Cognitive Annotation (Abend and Rappoport 2013)), as well as the development of open-domain semantic parsers that are able to produce formal semantic representations (e.g., Bos 2008, Butler 2015, Copestake and Flickinger 2000).

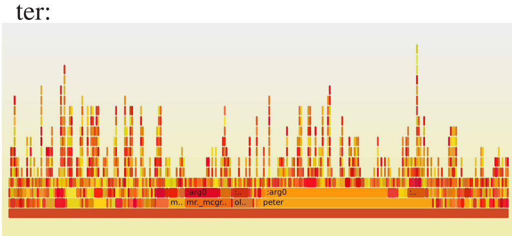
Open issues include how these different resources can be compared and their quality, coverage and accuracy fairly assessed. Also of concern is how the rich information of these resources can best be utilised, in particular for the extraction of content to feed tasks such as automatic summarisation and machine translation.

This paper considers flame graphs as a method of meaning presentation that is able to:

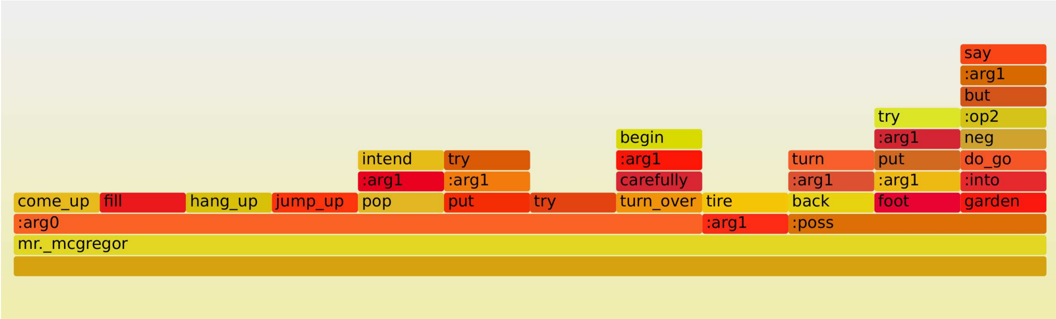
1. provide a normalised target for comparing, evaluating and error checking semantically annotated corpora and semantic parsing systems,
2. enable methods for extracting content, e.g., for summarisation, collecting frequency statistics, and
3. make visually accessible large volumes of semantic data to people without linguistic expertise who want to extract textual relations (e.g., Content Analytics in business/enterprise contexts; Zhu et al 2014).

2 An example

A flame graph consists of stacked rectangles. Each rectangle represents a distinct slice of semantic information. The wider a rectangle, the more often its content occurred. Colours of a graph are usually not significant, picked randomly to differentiate rectangles. The following is a graph for the 1,000 word children's story *The Tale of Peter Rabbit* by Beatrix Potter:



Lowermost on the y-axis, rectangles represent the entities of the story: a character such as Peter, Mr. McGregor, etc.; or an object such as a garden gate, a sieve, etc. The entities are sorted alphabetically to give the x-axis. Dependence on when in the story an entity is mentioned is lost in the graph with all information about an entity gathered above the same rectangle. From such arrangement it is clear to see, for example, that Peter is the principle character of the story, with 34.16% of what is said connected to Peter.



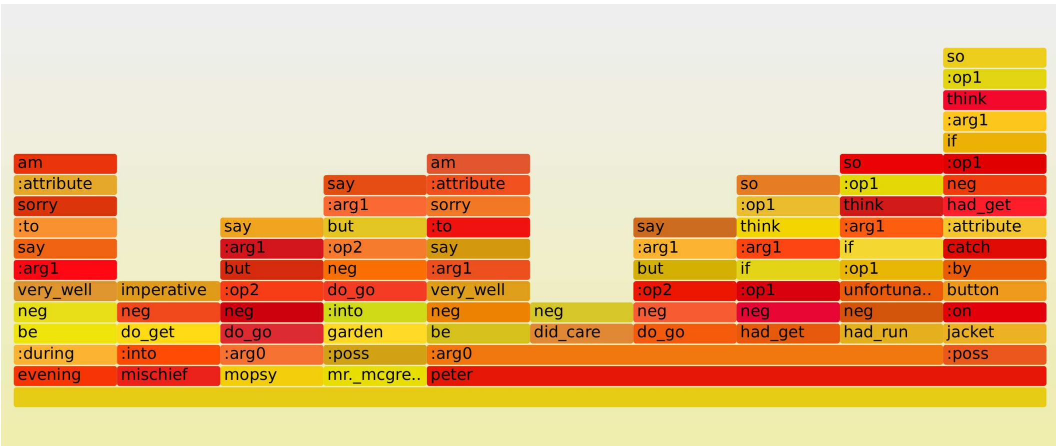
Examining a graph vertically reveals the semantic roles, predicates, argument roles, operators and connectives with scope over entities. For example, focus on information concerning Mr. McGregor, zooms to the above graph. This reveals that 66.67% of the information about Mr. McGregor concerns him doing things with '`arg0`' (logical subject) role. Other roles are '`arg1`' (logical object), and '`poss`' (possessive). Looking to the next row of rectangles gives the predicate relations for which the role holds, from which it can be seen that Mr. McGregor is: subject of *jumping up*, *popping*, etc; object of *tire*; possessor of ('`poss`' role) a *foot*, a *garden*, and so on.

If there are further rows above the predicate relations, then these rows provide information about the semantic context for the predicate relation. For example Mr. McGregor's *popping* falls under an '`arg1`' (complement) role which in turn falls under an *intending* predicate relation. It follows that while it can be concluded from the graph that Mr. McGregor does some *jumping up*, which occurs as a rectangle at the top of the graph, it cannot be concluded that he did any *popping*, which falls under the *intending* predicate relation. That is, it is the tops of the graph that are revealing the existential information of the story.

As a further example, the graph below focuses on negation occurring in the story. It can be seen that the majority of negation instances (8 of 10) occur at the fourth level, that is, as having immediate scope over the first predicate relation occurrence in the stack. In none of the cases in which negation occurs at a higher stack level does it scope over another scopal operator or connective, so all negations are narrow scoping in the Peter Rabbit story. The most complex case involves negation occurring at the tenth level but still inside the antecedent ('`op1`') of a conditional ('`if`') that is part of the '`arg1`' (complement) of *think* that is inside the antecedent ('`op1`') of a '`so`' coordination, stemming from the analysis of:

(1) After losing them, he ran on four legs and went faster, **so** that I **think** he might have got away altogether **if** he had **not** unfortunately run into a gooseberry net, and **got caught by** the **large buttons on his jacket**.

It also becomes a simple matter to read off statistics from the flame graph. For example, the flame graph for Peter Rabbit contains 163 distinct predicates, 112 of these appear more than once, 63 appear more than twice, and 28 appear five times or more.

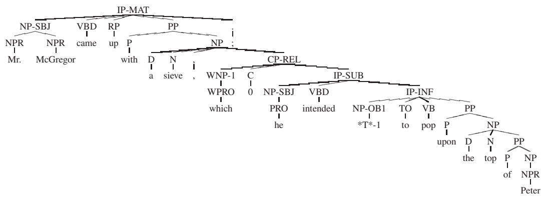


3 Building Flame Graphs

This section sketches how data for the above flame graphs was assembled using the Treebank Semantics method (`http://www.compling.jp/ts`; Butler 2015) of obtaining formal semantic representations from conventionally parsed constituent tree annotations. The first step required is to obtain a formal semantic analysis from a parsed syntactic tree. This is achieved by converting the syntactic tree to a formal language expression made up of instructions to manipulate the content of a sequence based information state (cf. Vermeulen 2000) to yield information to build a predicate logic based meaning representations as output. To see this with an example, consider (2).

(2) Mr. McGregor came up with a sieve, which he intended to pop upon the top of Peter;

First a parsing of (2) is required, e.g., the following tree representation that conforms to the *Annotation manual for the Penn Historical Corpora and the PCEEC* (Santorini 2010):

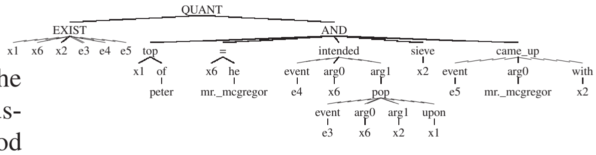


With conversion to a formal expression and subsequent processing the following meaning representation is returned:

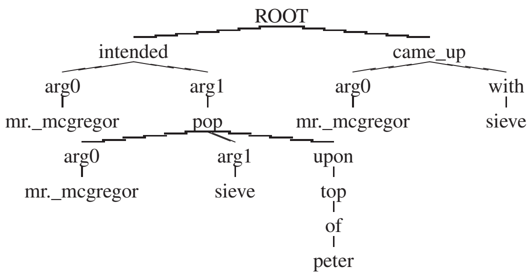
$$
\begin{array}{l}
\exists x_1 x_6 x_2 e_3 e_4 e_5 (\\
\quad \text{is\_top\_of}(x_1,\ \text{peter}) \wedge \\
\quad\quad x_6 = \text{he}\{\text{mr.\_mcgregor}\} \wedge \\
\quad\quad\quad \text{intended}(e_4,\ x_6, \\
\quad\quad\quad\quad \text{pop}(e_3,\ x_6,\ x_2) \wedge \\
\quad\quad\quad\quad\quad \text{upon}(e_3) = x_1) \wedge \\
\quad\quad\quad\quad\quad\quad \text{sieve}(x_2) \wedge \\
\quad\quad\quad\quad\quad\quad\quad \text{came\_up}(e_5,\ \text{mr.\_mcgregor}) \wedge \\
\quad\quad\quad\quad\quad\quad\quad\quad \text{with}(e_5) = x_2)
\end{array}
$$

This assumes a Davidsonian theory (Davidson 1967) in which verbs are encoded with minimally an implicit event argument which is existentially quantified over and may be further modified.

An alternative tree-based representation of the output meaning representation is as follows:



To reach information for a flame graph, transformations are made to eliminate bound variables by constructing "entity terms" from nominal predicates detectable as predicates with a bound argument that has no semantic role information. Thus `sieve` is used to replace all instances of `x2`, while `(top (of peter))` replaces instances of `x1`. Also, pronoun resolution has `mr._mcgregor` replace instances of `x6`. Remaining variables are eliminated from the representation, resulting in:



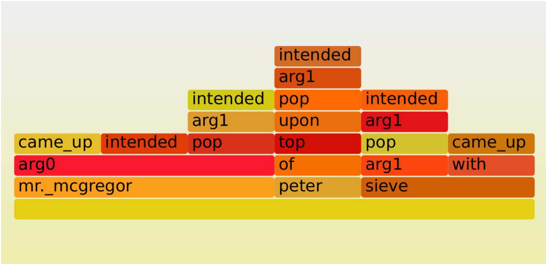
The above tree is processed to return all vertical slices:

```
ROOT intended arg0 mr._mcgregor
ROOT intended arg1 pop arg0 mr._mcgregor
ROOT intended arg1 pop arg1 sieve
ROOT intended arg1 pop upon top of peter
ROOT came_up arg0 mr._mcgregor
ROOT came_up with sieve
```

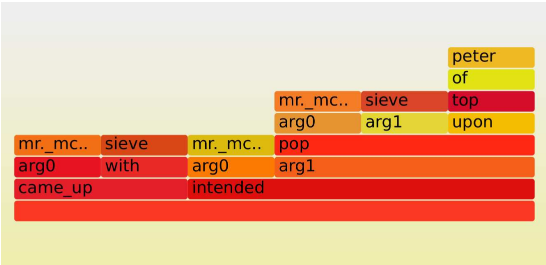
The content of each line is inverted and then lines are sorted alphabetically, with the resulting information suitably processed for the creation of a flame graph.

```
mr._mcgregor arg0 came_up
mr._mcgregor arg0 intended
mr._mcgregor arg0 pop arg1 intended
peter of top upon pop arg1 intended
sieve arg1 pop arg1 intended
sieve with came_up
```

The resulting flame graph is as follows:



When looking at the data of a single sentence, it is also instructive to look at the flame graph information reversed, that is, with entities as topmost elements of the graph.



4 Summary

To sum up, this paper has introduced a data visualisation technique called a flame graph that can be effective for charting information contained in large scale semantically annotated corpora or returned by open-domain semantic parsers. This has utility even when data size grows to thousands of sentences by supporting the ability to drill down from coarse to fine grained content. All semantic information is represented uniformly as stacked rectangles. Yet despite this simplicity, a great deal of semantic content is captured, with it being possible to distinguish mentioned entities, semantic / grammatical / thematic roles, properties, predicate relations, as well as connectives and operators with scope. Presentation is normalised to permit simple comparisons of form, e.g., offering evaluation metrics, as well as the ability to collect frequency statistics, e.g., to reveal trends in data for summarisation purposes.

References

Abend, Omri and Ari Rappoport. 2013. Universal conceptual cognitive annotation (ucca). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*.

Banarescu, L., C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

Basile, V., J. Bos, K. Evang, and N.J. Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the 8th Int. Conf. on Language Resources and Evaluation*. Istanbul, Turkey.

Bos, Johan. 2008. Wide-coverage semantic analysis with Boxer. In J. Bos and R. Delmonte, eds., *Semantics in Text Processing. STEP 2008 Conference Proceedings*, Research in Computational Semantics, pages 277–286. College Publications.

Butler, Alastair. 2015. *Linguistic Expressions and Semantic Processing*. Heidelberg: Springer.

Butler, Alastair and Kei Yoshimoto. 2012. Banking meaning representations from treebanks. *Linguistic Issues in Language Technology - LiLT* 7(1):1–22.

Copestake, Ann and Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000)*, page 591–600. Athens, Greece.

Davidson, Donald. 1967. The logical form of action sentences. In N. Rescher, ed., *The Logic of Decision and Action*. Pittsburgh: University of Pittsburgh Press.

Dowty, David, Robert Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*. Dordrecht: Kluwer.

Flickinger, Dan, Valia Kordoni, and Yi Zhang. 2012. Deepbank: A dynamically annotated treebank of the wall street journal. In *Proceedings of TLT-11*. Lisbon, Portugal.

Gregg, Brendan. 2013. Blazing performance with Flame Graphs. In *USENIX LISA 2013*. Washington, DC.

Santorini, Beatrice. 2010. Annotation manual for the Penn Historical Corpora and the PCEEC (Release 2). Tech. rep., Department of Computer and Information Science, University of Pennsylvania, Philadelphia.

Vermeulen, C. F. M. 2000. Variables as stacks: A case study in dynamic model theory. *Journal of Logic, Language and Information* 9:143–167.

Zhu, Wei-Dong, Bob Foyle, Daniel D Gagne, Vijay Gupta, Josemina Magdalen, Amarjeet S Mundi, Tetsuya Nasukawa, Mark Paulis, Jane Singer, and Martin Triska. 2014. *IBM Content Analytics Discovering Actionable Insight from Your Content, Third Edition*. ibm.com/redbooks.